# Improved Analysis of Time Series with Temporally Correlated Errors: An Algorithm that Reduces the Computation Time


*John Langbein*

*US Geologica Survey, Menlo Park, CA*

*langbein@usgs.gov*



ABSTRACT: Most time series of geophysical phenomena are contaminated with temporally correlated errors that limit the precision of any derived parameters. For example, estimates of station velocity derived from a time series of daily geodetic position measurements will be biased and its uncertainty will be too small if temporally correlated errors are ignored [Langbein and Johnson, 1997, Williams, 2003]. In particular, the rate uncertainty can be underestimated by a factor 10 for many GNSS time series. Obtaining better estimates of uncertainties is limited by several factors, including selection of the correct model for the background noise and the computational requirements to estimate the parameters of the selected noise model when there are numerous observations. Here, I address the second problem of computational efficiency using maximum likelihood estimates (MLE). Most geophysical time series have background noise processes that can be represented as a combination of white and power-law noise, $1/f^n$, with frequency, f. With missing data, standard spectral techniques involving Fourier transforms are not appropriate. Instead, time domain techniques involving construction and inversion of large data covariance matrices are employed. Bos et al. [2012] demonstrate one technique that substantially increases the efficiency of the MLE methods, but it provides only an approximate solution for power-law indices greater than 1.0 since they require the data covariance matrix to be Toeplitz. That restriction can be removed by simply forming a data-filter that adds noise processes rather than combining them in quadrature. Consequently, the inversion of the data covariance matrix is simplified and it provides robust results for a wide range of power-law indices. With the new formulation, the efficiency is typically improved by about a factor of 8 over previous MLE algorithms [Langbein, 2004].

The new algorithm can be downloaded at http://escweb.wr.usgs.gov/share/langbein/Web/OUT/est_noise/. The main program, est_noise7.2x, provides a number of basic functions that can be used to model the time-dependent part of time series, (rate, rate change, offset, exponential, Omori-law, sinusoids, and other, user supplied functions) and a variety of models that describe the temporal covariance of the data. These models of background noise include white noise, power-law noise, Gauss-Markov noise, and band-passed filtered noise; these can be combined to provide a complex mixture of noise processes. In addition, the new code provides a choice between adding the noise in quadrature, which has been the standard method, and summing the filter functions representing each noise process, which is the newer, faster method. Furthermore, the main program is packaged with a variety of utilities that can remove outliers, and importantly, help assess the success of the noise model with respect to the observations. These components are combined into example scripts which can help users analyze their own data.


**Background:** From GNSS time-series of positions, we extract velocities, accelerations, offsets, amplitudes of sinusoids, and post-seismic decay parameters, and importantly, the uncertainties of these values. This is accomplished using Least Squares regression. Key to estimating these parameters with any reasonable chance of accuracy is knowledge of the error structure of the time-series.

For computational simplicity, most investigators, at least in the past, assume that each observation of position is independent of its neighboring observation and that the error-model is Gaussian.

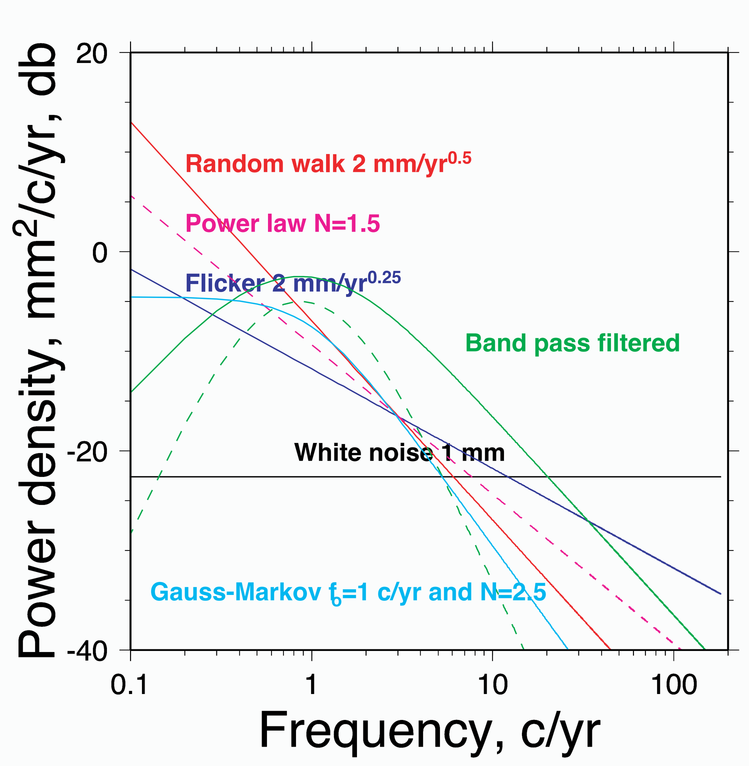
However, nearly all geophysical time-series have temporal correlations and these are quantified in Least Squares as the data covariance matrix.

For the assumption of no correlations, the covariance matrix is diagonal and the user will use "weighted Least Squares" to estimate the velocities, offsets, etc, along with their uncertainties.

However, in reality, the data are temporally correlated and that covariance is non-diagonal which complicates the regression: 1) one needs to "measure" the temporal covariance and 2) invert that large data covariance – Traditionally, both are computationally expensive.

LEAST SQUARES: $x=(A^tC^{-1}A)^{-1}A^tC^{-1}d$ where d is data, x are the model parameters (velocity, etc), A is the design matrix and $C^{-1}$ is the inverse of the data covariance.

Frequency domain representations of functions that can characterize temporal covariance (Langbein, 2004).




The most common method to simultaneously measure the temporal correlations and to estimate the functional model that describes the position time series (eg. velocity) is Maximum Likelihood Estimator (MLE). This relies upon making "guesses" of the parameters that describe a model for temporal correlations (which is typically a power-law in the frequency domain) and adjusting those parameters such that the likelihood is maximized. The algorithm of choice for optimization is the Nedler – Mead downhill simplex. This scheme has been used by Simon Williams in his CATS software and by me in est_noise6.50. Both of these algorithms handle time-series with missing observations or gaps. With proper optimization, typically using Cholesky decomposition to invert the data covariance, these codes can take at least 400 to 500 seconds to analyze 10 years of daily position GNSS data.

More recently Machiel Bos (*Bos et al.* 2012) developed a new algorithm, *Hector,* which, for time-series with only a few gaps, completes the MLE computation in less than 10 seconds for a 10 year long time series – Which is a significant improvement. The improved algorithm uses:

A new way of partitioning the data covariance – One part assumes that the data have no gaps and the second part represents the "missing" observations and

For the portion of the covariance matrix that models the complete data set, he makes the approximation that matrix is Toeplitiz and he has found an algorithm that can invert Toeplitz matrices efficiently.

PARTITIONING THE DATA COVARIANCE: $r^tC^{-1}r = r_o^t\{C_o^{-1} - C_o^{-1}M\,\underline{[M^tC_o^{-1}M]^{-1}}M^tC_o^{-1}\}r_o$

Inverse covariance with NO missing data --> Bos uses fast Toeplitz solver to invert, which is very fast!

Adjustment due to MISSING data; requires Cholesky decomposition which can be slow for many missing observations.

r is the difference between the observed and the predicted, $r_o$ is the same but for no missing data, and the combination of M and $M^t$ represent parts of $C_o^{-1}$ the with missing data.

**What is New**: Using the data covariance partitioning from Bos et al. (2012), I have found a method that removes the need for the Toeplitz approximation and yields similar computational efficiency as *Hector*. I call this new program *est_noise7.2x*.

Instead of adding in quadrature the constituent functional models that comprise data covariance, I create a single filter function by adding all of the constituent functional models that comprise the data covariance.

This filter function is used to create the non-missing portion of the data covariance. Significantly, by knowing that this is a filter function, it can be rapidly inverted either through deconvolution or inverting the discrete Fourier transform (DFT).

Traditional method of adding error in **quadrature**: $e^2 = e_1^2 + e_2^2 + e_3^2 + \ldots$ or $C = C_1 + C_2 + C_3 + \ldots$ where $e_i$ and $C_i$ are comprised of a single source of error such as white noise or power law noise. Or, $e_i = f_i * w_i$ where the filter, $f_i$, is convolved with an independent source of white noise, $w_i$. $f_i$ represents one of the noise functions shown to the left.

An alternative composition of error is to **add** the desired noise functions, shown to the left, then convolve that more complex filter with a single source of white noise; $e = [f_1 + f_2 + f_3 + \ldots] * w$

Consequently, the inverse filter function is easily solved by deconvolution of: □ $= f_t * f_t^{-1}$, where □ is the delta function. Therefore, the covariance matrix is $C=F\,F^t$, and its inverse is $C^{-1} = F^{-1t}F^{-1}$. Like Bos et al.'s Toeplitz solver, the inverse calculation is *fast* and *without* the restriction of the covariance matrix being Toeplitz.

*The key difference between the traditional quadrature and the additive method proposed here is the order of adding and squaring; Quadrature Squares then Adds; my Additive method Adds then Squares.*

## Comparison of three algorithms

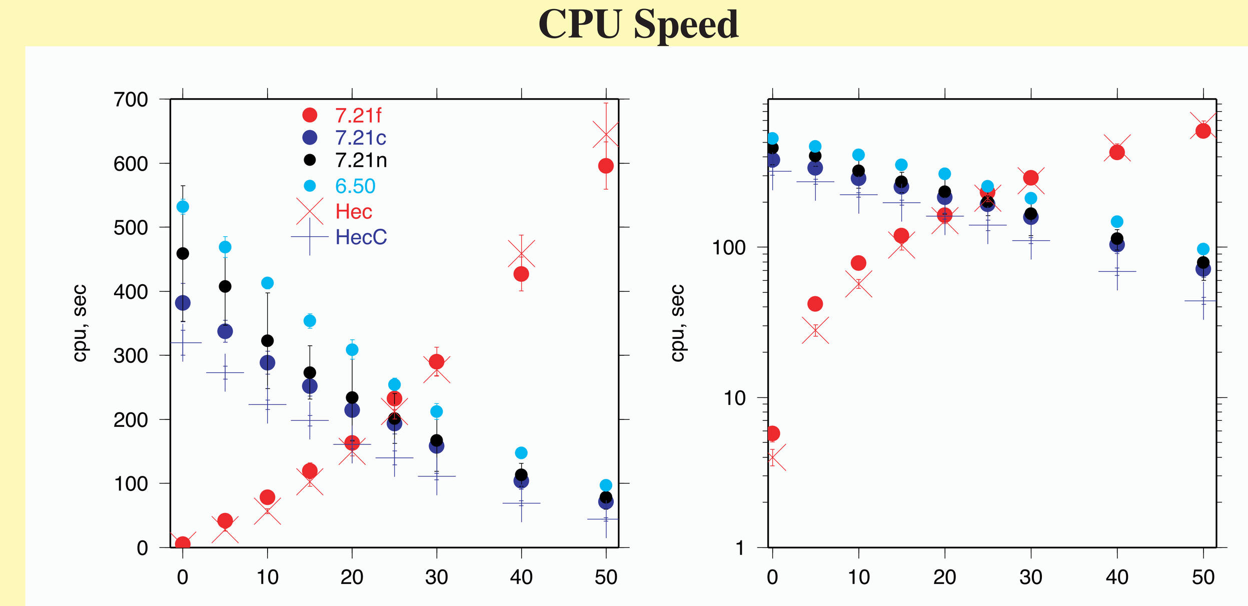
| Program | est_noise6.50 | est_noise7.2x | | | hector | |
|---|---|---|---|---|---|---|
| ID | 6.50 | 7.21n | 7.21c | 7.21f | HecC | Hec |
| Data Error | quad | quad | additive | additive | quad | quad |
| Toeplitz approx. | no | no | no | no | yes | yes |
| Inverse routine | Cholesky | Chol. | Chol. | Bos | Chol. | Bos |

Bos et al. (2012) covariance partitioning

**Simulations:** To compare these programs and options, I created 15 sets of data having a combination of 0.7 mm of white noise and 3.0 mm/yr$^{0.375}$ of power law noise with an index of 1.5. Each data set originally had 4000 observations with none missing. To test the performance of each program and its options, I ran several tests using each program and its options and I recorded the speed of computation (cpu speed), the estimates of the white and power-law parameters, along with the rate, size of offset, and their errors. Furthermore, for each set of simulated data, I arbitrarily remove 5, 10, 15, 20, 25, 30, 40 and 50% of the time series and re-ran all six options. (Note; the power law index for these simulations was chosen to be 1.5 because the version of *Hector* that I have has a limit of 1.6 due to the Toeplitz solver/approximation used in the covariance inversion.)

## Algorithm Performance; Comparison

### CPU Speed

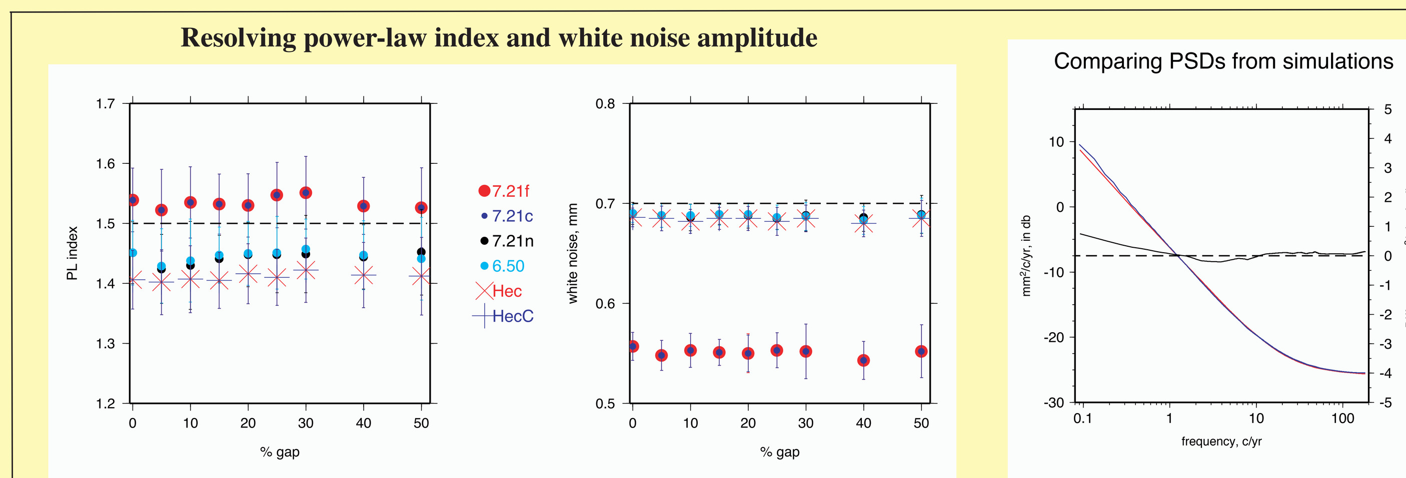



With no gaps in the data, both algorithms (Hec and 7.21f) that employ the covariance partitioning complete the analysis in 5 to 8 seconds, or about 50 to 100x faster than the algorithms that use Cholesky decomposition.

However, with more gaps in the data, the cpu time increases for Hec and 7.21f because it becomes more expensive to do the Cholesky decomposition of the right hand side of the partitioned covariance.

On the other hand, with more gaps in the data, size of the covariance matrix decreases and Cholesky decomposition is more efficient relative to the partitioning method.

**Resolving power-law index and white noise amplitude**




The actual estimates of the parameters describing the noise model depend upon how the data covariance is constructed: quadrature or addition of the filter functions.

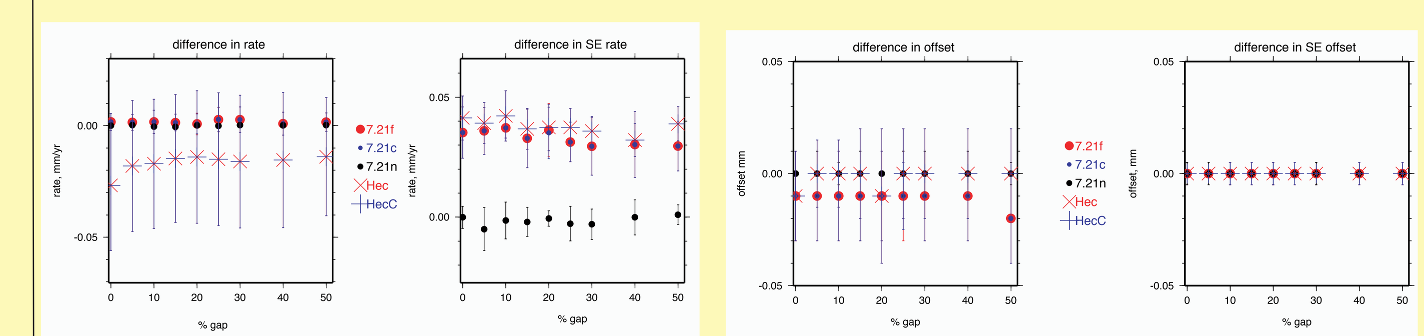
*Yet, independent of the construction of the covariance, the power spectra for both approaches are the same!*

**Comparison of the estimate of rate, offset, and their standard errors**

Simulated rate is 0.00+/-0.24 mm/yr

Simulated offset is 0.00+/-0.49 mm/yr

Plots show **differences** relative to the "legacy" MLE program, 6.50




The differences between the estimates of rates, offsets, and their uncertainties are relatively minor independent of the algorithm and the construction of the data covariance.

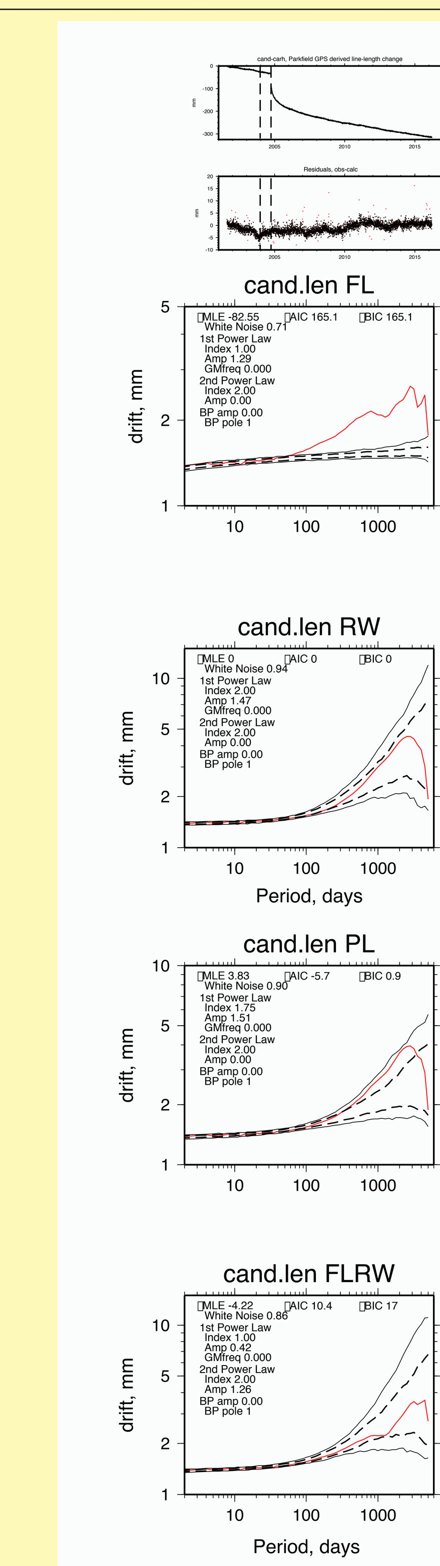
## Example

**GNSS data** from Parkfield showing the displacements from the San Simeon and Parkfield Earthquakes.

**Residuals** after removing a rate, co-seismic offsets, and post-seismic deformation (Omori law). Outliers are shown in red and are removed from time-series prior to analysis of noise properties and time-dependent deformation.

**Four different models** of noise have been analyzed by *est_noise7.2x*. One can evaluate the success of each model by examining several coefficients including the Maximum Likelihood Estimator (MLE), and the Akaike/Bayesian Information Criterion (AIC/BIC), or graphically using "wander" or "drift". Drift, d(□) is defined as RMS of [x(t+□)-x(t)] as a function of various intervals, □, for time series, x(t). The drift of the data residuals can be graphically compared with simulated data having the same noise as the real data.

Since flicker or random-walk plus white noise is the simplest variety of colored noise, the one that maximizes the likelihood is termed the Null model. For this Parkfield example, RW + WN is the better of the two and is designated as Null. Adding complexity with power-law, flicker plus random-walk, Gauss-Markov, and band-passed filtered noise improves the MLE, but in terms of the graphical drift and AIC/BIC coefficients, indicates that additional complexity is not required; hence these data are well characterized by random-walk noise.




## Conclusion:

New algorithm provides up to ***50X*** improvement in cpu speed over older version of MLE codes that measure temporal correlations of the and simultaneously fit time-dependent functions to the data.

More typical performance is closer to ***10X*** for most GNSS time series.


**References:**

Bos, M.S., R.M.S. Fernandes, S.P.D. Williams, and L. Bastos, Fast error analysis of continuous GNSS observations with missing data, J. Geod., doi:10.1007/s00190-012-0605-0 (2012).

Langbein, J., and H. Johnson, Correlated error in geodetic time series: Implications for time-dependent deformation, J. Geophy. Res., 102, 591-604 (1997).

Langbein, J., Noise in two-color electronic distance meter measurements revisited, J. Geophy. Res., doi:10.1029/2003JB002819 (2004).

Williams, S.D.P, The effect of coloured noise on the uncertainties of rates estimated from geodetic time series., J. Geod, doi:10.1007/s00190-002-0283-4, (2003).